

VISUAL INSTRUCTION SET FOR CPU HAVING INTEGRATED GRAPHICS FUNCTION

Publication number: JP11003226 (A)

Publication date: 1999-01-06

Inventor(s): YUNG ROBERT

Applicant(s): SUN MICROSYSTEMS INC

Classification:

- **international:** *B41J5/30; G06F7/544; G06F9/30; G06F9/302; G06F9/38; G06T1/20; B41J5/30; G06F7/48; G06F9/30; G06F9/302; G06F9/38; G06T1/20; (IPC1-7): G06F9/38; B41J5/30; G06F9/30; G06T1/00*

- **European:** G06F7/544A; G06F9/302

Application number: JP19970278845 19971013

Priority number(s): US19960722442 19961010

Also published as:

EP0836137 (A2)

EP0836137 (A3)

US5996066 (A)

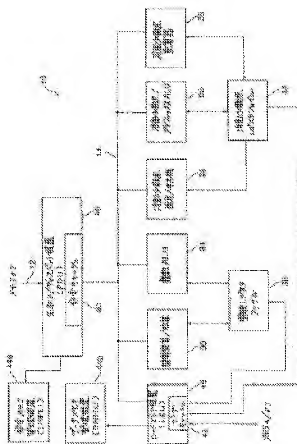
SG103284 (A1)

SG54569 (A1)

Abstract of JP 11003226 (A)

PROBLEM TO BE SOLVED: To quickly execute other graphics instructions by using a special instruction to minimize the addition of hardware to a general-purpose CPU and also to optimize the execution of a certain graphics instruction.

SOLUTION: A PDU(prefetch dispatch unit) 46 gives four instructions at most and in parallel to plural pipeline type execution devices along a pipeline bus 14. An integer calculation instruction is sent to one of two IEU(integer execution units), i.e., an integer multiplication/division unit 30 or an integer ALU (arithmetic logic unit) 31. These two units share the access to an integer register file 36 to store the operand and result of the integer calculation.; In addition, three floating point arithmetic units, i.e., a floating point division/ square root execution unit 25, a floating point/graphics ALU 26 and a floating point/graphics multiplier 28 are connected to the bus 14 and share a floating point register file 38.



Data supplied from the **esp@cenet** database — Worldwide

特開平11-3226

(43) 公開日 平成11年(1999) 1月6日

(51) Int.Cl. ⁶	識別記号	F I
G 0 6 F 9/38	3 7 0	C 0 6 F 9/38 3 7 0 X
B 4 1 J 5/30		B 4 1 J 5/30 Z
G 0 6 F 9/30	3 5 0	C 0 6 F 9/30 3 5 0 A
G 0 6 T 1/00		15/66 J

審査請求 未請求 請求項の数46 O L (全 25 頁)

(21) 出願番号	特願平9-278845	(71) 出願人	591064003 サン・マイクロシステムズ・インコーポレ ーテッド SUN MICROSYSTEMS, IN CORPORATED アメリカ合衆国 94303 カリフォルニア 州・パロ アルト・サン アントニオ ロ ード・901
(22) 出願日	平成9年(1997)10月13日	(72) 発明者	ロバート・ユング アメリカ合衆国・94556・カリフォルニア 州・フレモント・コマース ドライブ・ 5797
(31) 優先権主張番号	0 8 / 7 2 2 4 4 2	(74) 代理人	弁理士 山川 政樹
(32) 優先日	1996年10月10日		
(33) 優先権主張国	米国 (US)		

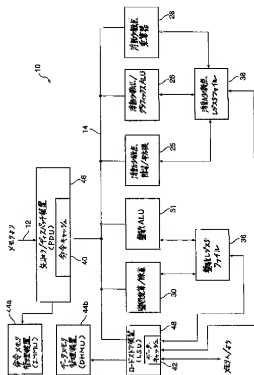
最終頁に続く

(54) 【発明の名称】 統合グラフィックス機能を有するCPU用のビジュアル命令セット

(57) 【要約】

【課題】標準的なマイクロプロセッサの整数／浮動小数点演算に加えてグラフィックス演算をサポートするための最適化されたスーパー scaler・マイクロプロセッサアーキテクチャを提供する。

【解決手段】汎用CPUへのハードウェアの追加を最小限にしてグラフィックス命令の実行を最適化するためのいくつかの特殊なグラフィックス命令及びそれらの命令を実行するためのハードウェアである。



【特許請求の範囲】

【請求項1】 グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサにおいて：第1のソースレジスタと；第2のソースレジスタと；宛先レジスタと；前記3つのレジスタの中の2つのレジスタに接続された第1及び第2の入力を有し、乗算／加算演算コードにตอบสนองしてその2つのレジスタの各レジスタの中の複数の値に対して区分乗算を実行する乗算器論理回路と；前記3つのレジスタの中の第3のレジスタに接続された第1の入力及び乗算器論理回路の出力に接続された第2の入力を有し、乗算／加算演算コードにตอบสนองして第3のレジスタの中の複数の値と乗算器から出力される複数の値との区分加算を実行する加算器と；を具備したマイクロプロセッサ。

【請求項2】 前記3つのレジスタの中の少なくとも1つのレジスタのどの区分フィールドに対して演算を加えるかを指示するマスクレジスタをさらに具備した請求項1記載のマイクロプロセッサ。

【請求項3】 グラフィックス演算及びノングラフィックス演算の両方演算を実行するマイクロプロセッサにおいて：第1のソースレジスタと；第2のソースレジスタと；宛先レジスタと；前記3つのレジスタの中の2つのレジスタに接続された第1及び第2の入力を有し、乗算／減算演算コードにตอบสนองしてその2つのレジスタの各レジスタの中の複数の値に対して区分乗算を実行するよう構成された乗算器論理回路と；前記3つのレジスタの中の第3のレジスタに接続された第1の入力及び乗算器論理回路の出力に接続された第2の入力を有し、乗算／減算演算コードにตอบสนองして第3のレジスタの中の複数の値と乗算器から出力される複数の値との区分減算を実行するよう構成された加算器と；を具備したマイクロプロセッサ。

【請求項4】 前記3つのレジスタの中の少なくとも1つのレジスタのどの区分フィールドに対して演算を加えるかを指示するマスクレジスタをさらに具備した請求項3記載のマイクロプロセッサ。

【請求項5】 グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサにおいて：マイクロプロセッサに第1のレジスタにバックされた複数の第1のレジスタ値と第2のレジスタにバックされた複数の第2のレジスタ値との区分乗算を実行させて複数の乗算結果を生じさせると共に、その乗算結果と第3のレジスタにバックされた複数の第3のレジスタ値との区分加算を実行させる演算コード命令を具備したメモリ。

【請求項6】 前記3つのレジスタの中の少なくとも1つのレジスタのどの区分フィールドに対して演算を加えるかを指示するマスクをセットするための演算コード命令をさらに具備した請求項5記載のメモリ。

【請求項7】 グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサによってアクセス可能なコンピュータ読み取り可能なメモリにおいて：マイクロプロセッサに第1のレジスタにバックされた複数の第1のレジスタ値と第2のレジスタにバックされた複数の第2のレジスタ値との区分乗算を実行させて複数の乗算結果を生じさせると共に、その乗算結果と第3のレジスタにバックされた複数の第3のレジスタ値との区分減算を実行させるよう構成された演算コード命令を具備したメモリ。

【請求項8】 前記3つのレジスタの中の少なくとも1つのレジスタのどの区分フィールドに対して演算を加えるかを指示するマスクをセットするための演算コード命令をさらに具備した請求項7記載のメモリ。

【請求項9】 グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサにおいて：ソースレジスタと；ソースレジスタに接続された入力を有し、ソースレジスタ中の複数の各値の平方根分の1の値を並列に計算するよう構成された除算・平方根論理回路と；を具備したマイクロプロセッサ。

【請求項10】 前記除算・平方根論理回路がルックアップテーブルを具備する請求項9記載のマイクロプロセッサ。

【請求項11】 前記除算・平方根論理回路が反復論理回路を具備する請求項9記載のマイクロプロセッサ。

【請求項12】 グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサによってアクセス可能なコンピュータ読み取り可能なメモリにおいて：入力ソースレジスタにおける複数の各区分フィールドの値の平方根分の1の値の計算を並列に実行させるようマイクロプロセッサに命じる演算コードを具備したメモリ。

【請求項13】 グラフィックス演算及びノングラフィックス演算の両方の演算を実行するためのマイクロプロセッサにおいて：複数の区分フィールドを有するソースレジスタと；宛先レジスタと；マスクレジスタと；ソースレジスタと宛先レジスタとの間に接続されていて、抽出命令にตอบสนองしてソースレジスタから宛先レジスタへ、マスクレジスタによって決定される区分フィールドの中の選択された区分フィールドを送って記憶させるよう構成された論理回路を具備したマイクロプロセッサ。

【請求項14】 前記論理回路が、選択された区分フィールドを宛先レジスタの最下位フィールドに記憶させるように構成されている請求項13記載のマイクロプロセッサ。

【請求項15】 前記論理回路が、選択された区分フィールドを宛先レジスタの対応するフィールドに記憶させて、ソースレジスタと宛先レジスタの内容のマージを実行させるよう構成されている請求項13記載のマイクロプロセッサ。

【請求項16】グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサによってアクセス可能なコンピュータ読み取り可能なメモリにおいて、指定された値をマスクレジスタに入力させるようにマイクロプロセッサに命じる第1の命令と；ソースレジスタから宛先レジスタへ、マスクレジスタによって決定される区分フィールド中の選択された区分フィールドを送って記憶させるようにマイクロプロセッサに命じる第2の命令と；を具備したメモリ。

【請求項17】前記選択された区分フィールドが宛先レジスタの最下位フィールドに記憶される請求項16記載のメモリ。

【請求項18】前記選択された区分フィールドが宛先レジスタの対応するフィールドに記憶されて、ソースレジスタと宛先レジスタの内容のマージを実行させる請求項16記載のメモリ。

【請求項19】グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサにおいて、複数の区分フィールドを有するソースレジスタと；宛先レジスタと；ソースレジスタに接続されていて、指定された種類の先行ディジットまたはディジット列の場所を検出すると共に、先行ディジットに対するポインタを宛先レジスタに記憶する検出論理回路と；を具備したマイクロプロセッサ。

【請求項20】前記指定された種類の先行ディジットが1である請求項19記載のマイクロプロセッサ。

【請求項21】前記指定された宛先ディジットがゼロである請求項19記載のマイクロプロセッサ。

【請求項22】前記検出論理回路が優先順位デコーダを具備する請求項19記載のマイクロプロセッサ。

【請求項23】前記検出論理回路がシフトレジスタを具備する請求項19記載のマイクロプロセッサ。

【請求項24】グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサによってアクセス可能なコンピュータ読み取り可能なメモリにおいて、ソースレジスタ中の指定された種類の先行ディジットまたはディジット列の場所を検出させると共に、宛先ディジットに対するポインタを宛先レジスタに記憶させるようマイクロプロセッサに命じる命令を具備したメモリ。

【請求項25】前記ポインタが最下位ビットからのオフセットである請求項24記載のメモリ。

【請求項26】グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサにおいて、整数レジスタファイルと；浮動小数点・グラフィックスレジスタファイルと；浮動小数点・グラフィックスレジスタファイルのレジスタの内容を整数レジスタファイルのレジスタに移動させる交換論理回路と；を具備したマイクロプロセッサ。

【請求項27】グラフィックス演算及びノングラフィッ

クス演算の両方の演算を実行するマイクロプロセッサによってアクセス可能なコンピュータ読み取り可能なメモリにおいて、浮動小数点・グラフィックスレジスタファイルのレジスタの内容を整数レジスタファイルのレジスタにへ移動させるようマイクロプロセッサに命じる命令を具備したメモリ。

【請求項28】グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサにおいて、複数の区分フィールドを有するソースレジスタと；ソースレジスタに接続されていて、各区分フィールドのビットを隣接する区分フィールド中にシフトさせずにシフトさせるシフト論理回路と；シフトオペレーンで使用される少なくとも1ビットを記憶する制御レジスタと；を具備したマイクロプロセッサ。

【請求項29】前記シフト論理回路が、少なくとも1つの区分フィールドからのビットを制御レジスタにシフトさせるよう構成されている請求項28記載のマイクロプロセッサ。

【請求項30】前記制御レジスタが、区分フィールドの中のどの区分フィールドをシフトさせるかを決定するためのマスクレジスタを具備する請求項28記載のマイクロプロセッサ。

【請求項31】前記シフト論理回路が、左シフト命令に応答して、ビットをゼロを最下位ビット位置に加えて左シフトさせるよう構成されている請求項28記載のマイクロプロセッサ。

【請求項32】前記シフト論理回路が、右シフト命令に応答して、ビットを符号ビットを各区分フィールドの最上位ビット位置にコピーして右シフトさせるよう構成されている請求項28記載のマイクロプロセッサ。

【請求項33】前記シフト論理回路が、右シフト命令に応答して、ビットをゼロを各区分フィールドの最上位ビット位置に加えて右シフトさせるよう構成されている請求項28記載のマイクロプロセッサ。

【請求項34】グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサによってアクセス可能なコンピュータ読み取り可能なメモリにおいて、複数の各区分フィールドのビットを隣接した区分フィールド中にシフトさせずにシフトさせ、シフトに使用される少なくとも1ビットを制御レジスタに記憶するようにマイクロプロセッサに命じる命令を具備したメモリ。

【請求項35】前記命令が、区分フィールドの中の少なくとも1つの区分フィールドから制御レジスタへビットをシフトさせるよう構成されている請求項34記載のメモリ。

【請求項36】前記区分フィールドの中のどの区分フィールドをシフトさせるかを決定するためのマスクレジスタにマスクを書き込むための命令をさらに具備した請求項34記載のメモリ。

【請求項37】グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサにおいて、ソース記憶場所と；宛先レジスタと；マスクレジスタと；レジスタファイルとマスクレジスタに接続されていて、マスクレジスタによって決定される区分フィールドの中の選択された一群の区分フィールドをソースレジスタから宛先レジスタへ移動させる移動論理回路と；を具備したマイクロプロセッサ。

【請求項38】前記選択された一群の区分フィールドに対して指定された演算を実行するよう構成された実行論理回路をさらに具備した請求項37記載のマイクロプロセッサ。

【請求項39】前記ソース記憶場所がソースレジスタである請求項37記載のマイクロプロセッサ。

【請求項40】グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサによってアクセス可能なコンピュータ読み取り可能なメモリにおいて、指定された値をマスクレジスタに入力させるようにマイクロプロセッサに命じる第1の命令と；ソースレジスタから宛先レジスタへ、マスクレジスタに従って決定される区分フィールドの中の選択された一群の区分フィールドを移動させるようにマイクロプロセッサに命じる第2の命令と；を具備したメモリ。

【請求項41】グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサにおいて、アドレスレジスタ；アドレスレジスタに接続された加算器と；グラフィックスデータ宛先レジスタと；アドレスレジスタ及び加算器に接続されていて、アドレスレジスタ中のアドレスによって指示されるメモリ内のアドレスのグラフィックスデータを宛先レジスタの中にロードすると共に、加算器を用いてアドレスレジスタを修正するよう構成されている制御論理回路と；を具備したマイクロプロセッサ。

【請求項42】前記制御論理回路が、データサイズに従ってアドレスレジスタをインクリメント、またはデクリメントさせるよう構成されている請求項41記載のマイクロプロセッサ。

【請求項43】グラフィックス演算及びノングラフィックス演算の両方の演算を実行するマイクロプロセッサによってアクセス可能なコンピュータ読み取り可能なメモリにおいて、アドレスレジスタ中のアドレスによって指示されるメモリ内のアドレスのグラフィックスデータを宛先レジスタの中にロードすると共に、データサイズを用いてアドレスレジスタを修正するようにマイクロプロセッサに命ずる命令を具備したマイクロプロセッサ。

【請求項44】前記データサイズをデータサイズレジスタに入力させるようにマイクロプロセッサに命ずる第2の命令をさらに具備した請求項43記載のメモリ。

【請求項45】前記乗算及び加算の結果を丸め、中間結

果は丸めない丸め論理回路をさらに具備した請求項1記載のマイクロプロセッサ。

【請求項46】前記乗算及び減算の結果を丸め、中間結果は丸めない丸め論理回路をさらに具備した請求項3記載のマイクロプロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、統合グラフィックス機能を有するスーパースーパー中央処理装置（CPU）に関するものである。

【0002】

【従来の技術】歴史的に見ると、初期の頃の従来技術のコンピュータシステムにおけるCPUはグラフィックスとノングラフィックスの両方の機能を扱って来た。それからしばらく後の従来技術のコンピュータシステムでは、補助ディスプレイプロセッサが用いられた。それより後の他の従来技術によるコンピュータシステムでは、たとえば補助グラフィックスプロセッサが用いられるようになった。これらのグラフィックスプロセッサは、汎用CPUの場合における図形処理の大半の部分を行うものであった。

【0003】マイクロプロセッサの場合は、技術の進歩によって、小さな面積内にますます多くの電気回路を実装することができるようになるにつれて、前記のようなグラフィックスプロセッサの代わりに、汎用CPUにビルトイングラフィックス機能機能を一体化して組み込むことがしだいに多く望まれるようになってきた。従来技術による一部の新しいコンピュータシステムでは、このような機能の組み込みが始まっている。しかしながら、これらの新しい従来技術のコンピュータシステムに統合して組み込まれるグラフィックス機能の量と性質は、まだ非常に限定されており、トレードオフを伴うのが普通である。CPUに統合して組み込まれていることが知られている具体的なグラフィックス機能としては、フレームバッファチェック、ピクセルマージ付き加算、及びzバッファマージ付き加算がある。これらの新しい従来技術のシステム上で行われる図形処理の大部分は、依然としてビルトイングラフィックス機能が付加されていない汎用CPUにより、あるいは補助ディスプレイ/グラフィックスプロセッサによって処理されている。

【0004】グラフィックス機能を組み込んだRISCマイクロプロセッサの一例にモトローラMC88110がある。このマイクロプロセッサは、その整数実行装置、乗算、除算及び浮動小数点加算装置の他に、2つの専用グラフィックス装置が付加されている。その付加グラフィックス装置は、ピクセル加算実行装置及びピクセルバック実行装置である。このモトローラプロセッサでは、他の実行装置で他の機能のために使用される64ビットのデータ経路に複数のピクセルをバックすることが可能である。従って、複数のピクセルを一度に処理する

ことができる。バックギン実行装置におけるバックギンオペレーションは、ピクセルを64ビットフォーマットにバックする。ピクセル加算オペレーションでは、相互にピクセル値を足したり、引いたりすることが可能であり、複数のピクセルが64ビットフィールドで一度に減算される。これには、各各ビット境界毎に通常加算器で生じる桁上げをディスイープルにする必要がある。また、このモトローラプロセスは、乗算結果が次のピクセル値表現中に溢れ出ないように、ピクセルをゼロが高位ビットに入るようにしてフィールドに入れて通常の乗算装置を用いて行われるピクセル乗算オペレーションも可能なようになっている。

【0005】インテル 1860マイクロプロセッサは、Zバッファ・グラフィックス命令の実行を可能にするグラフィックス装置を組み込んだものである。これらは、基本的には、3次元ディスプレイでどのピクセルを他のピクセル前に置くべきであるか決定するために必要な複合オペレーションである。インテルMMX命令セットは、モトローラMC88110で提供される命令を拡張した形で、汎用マイクロプロセッサ上で実行される多数の区分グラフィックス命令を与える。

【0006】

【発明が解決しようとする課題】複数のピクセル値を有するバック型区分レジスタを用いて他のグラフィックス機能をより迅速に実行できるようにすることが要望されている。

【0007】

【課題を解決するための手段】本発明は、標準的なマイクロプロセッサの整数及び浮動小数点演算に加えて、グラフィックス演算をサポートするための最適化されたスーパースカラー・マイクロプロセッサアーキテクチャである。本願では、汎用CPUへのハードウェアの追加を最小限にしてグラフィックス命令の実行を最適化するためのいくつかの特殊なグラフィックス命令及びそれらの命令を実行するためのハードウェアが開示される。本発明においては、グラフィックス演算でしばしば必要になる特殊な論理演算を行うことが可能である。特に、一つの命令がオペランドの平方根で割った1の値を計算し、もう一つの命令が、マスク機能によって、2つの区分値の掛算及びこれらの値とは別の第3の値との加算を両方とも行うことが可能である。これらの各命令は、1つのレジスタ中の複数の区分ピクセル値に対して作用する。

【0008】あちこちの区分ピクセルフィールドを処理するためにいくつかの命令が用意されている。特に、抽出オペレーションによってソースレジスタの指定されたフィールド宛先レジスタに格納することが可能になる。または、指定されたビットを抽出することもできる。指定されたフィールドまたはビットをマスクレジスタによって指示することができる。さらに、マスクレジスタを用いて条件付き移動、ロードまたは実行を行うことによ

り、どの区分フィールドまたはビットについて処理を行うかを指示することができる。

【0009】もう一つの命令が、先行1または先行ゼロを検出して、その位置にボインズを戻す。あるいは、ストリング探索を用いて特定のパターンを検出するようにしてもよい。これは、暗号化及びデータ圧縮/圧縮解除に役立つ。

【0010】もう一つの特殊な命令は、浮動小数点レジスタファイルと整数レジスタファイルとの間のアドレスまたはデータの交換を可能にするものである。もう一つの命令は、マスクがどのフィールドをシフトさせるかを指定するか、あるいは1つ以上のフィールドからシフトアウトされるビットを記憶するようにして、複数の区分フィールドが内部の区分フィールド中にシフトすることなくそれぞれ連続して並列にシフトされるようにしたマスクによる区分送りを行う可能にする。

【0011】また、本発明は、記憶場所からグラフィックスレジスタへのロードも可能で、この場合ロードオペレーションはアドレスレジスタをインクリメントする。さらに、本発明は複数の区分変数について、変数の絶対値を変数自身に加えるための命令を提供する。

【0012】また、本発明によれば、単一命令による区分除法演算が可能である。

【0013】以下、添付図面を参照して本発明をさらに詳細に説明する。

【0014】

【発明の実施の形態】

CPUの全体的アーキテクチャ

図1は、本発明の技術内容を組み込んだ一例のグラフィックスコンピュータシステムのCPUを図解したブロック図である。

【0015】図示のように、CPU10は命令キャッシュ40に接続された先取り/ディスペッチ装置(PDU)46を有する。命令は、命令メモリ管理装置(LMMU)44aを利用してバス12上のキャッシュまたは主記憶装置からの装置によってフェッチされる。データは、データメモリ管理装置(DMMU)44bと共に動作するロード記憶装置(LSU)48を用いて主記憶装置またはデータキャッシュ42からフェッチされる。

【0016】PDU46はパイプラインバス14に沿って複数のパイプライン式実行装置に並列に最大4つの命令を出す。整数演算命令は2つの整数実行装置(IEU)、すなわち整数乗算/除算装置30及び整数ALU31の1つに送られる。これらの2つの装置は、整数レジスタファイル36へのアクセスを共有することによって整数演算のオペランド及び結果を記憶する。

【0017】これらとは別に3つの浮動小数点演算装置が備わっている。すなわち、浮動小数点除算/平方根実行装置25、浮動小数点/グラフィックスALU26、浮動小数点/グラフィックス乗算器28がパイプライン

バス14に接続されており、浮動小数点レジスタファイル38を共有する。浮動小数点レジスタファイルは、浮動小数点演算及びグラフィックス演算のオペランドと結果とを記憶する。

【0018】浮動小数点装置26及び28を通るデータ経路は、並列に8〜8ビットピクセル表現（あるいは4〜16ビット、または2〜32ビット表現）に対応することができるよう64ビットに拡張されている。このように、53ビット+3エキストラビット（保護ビット、丸めビット及びステッキビット、すなわちGRS）の標準的浮動小数点経路は、本発明によりグラフィックス命令に対応するよう拡張されている。本発明は任意のデータサイズに適用することができる。直列の複数の64ビット量に対して命令を作用させることにより、あるいはより大きいレジスタ及びバスサイズを用いることによって64ビットのレジスタサイズ及び演算サイズを使用することもできる。

【0019】さらに、IEUはいくつかのグラフィックス演算も行い、LSD48用のロード／ストア命令のアドレスにアドレス空間識別子（ASI）を付加して、アクセス中のアドレス空間を識別する。また、LSU48は特にグラフィックスデータのために設計されたいくつかのロード及びストアオペレーションをサポートする。メモリ参照は仮想アドレスで行われる。MMU44a〜44bは、仮想アドレスを物理アドレスにマップするためのアドレス変換案内バッチ（TLB）を有する。

【0020】2つの区分グラフィックス実行経路図2は、浮動小数点／グラフィックス実行装置26及び28をより詳細に示したものである。図2は、これらの実行装置によってグラフィックス命令のための2つの区分実行経路、すなわち装置26による第1の区分実行経路と装置28による第2の区分実行経路が得られることを示している。これらの両経路は、先取り／ディスパッチ装置46に接続されたパイプラインバス14に接続されている。ハードウェアと命令を2つの異なる実行経路に分けることによって、パイプラインの各々のサイクル毎に2つの独立したグラフィックス命令を並列に実行することができる。このように命令及びハードウェアを2つの経路に分けることは、典型的なグラフィックスアプリケーションのスループットを最適化するために必要なものである。

【0021】図2には、グラフィックス状態レジスタ（GSR）50も示されている。このレジスタは、両方の実行経路でグラフィックス命令により使用されるスケールファクタ及び整列オフセットデータを記憶するので、2つの経路の外部に設けられている。各実行経路は、バス18を介してグラフィックス状態レジスタ中の情報を供給される。グラフィックス状態レジスタは、バス20を介してIEUにより書き込まれる。

【0022】グラフィックス状態レジスタ

次に、図3には、グラフィックス状態レジスタ（GSR）の一実施形態の関連部分が図解されている。この実施形態においては、GSR50はビット0〜2にオフセットを、ビット3〜8にスケールファクタを記憶するために使用され、残りのビットは予備として残される。オフセットは、整列前のピクセルアドレスの最下位3ビット（alignaddr_offset）54であり、スケールファクタ52はピクセルフォーマット化のために使用される。alignaddrオフセット54はGSR[2:0]ビットに記憶され、スケールファクタ52はGSR[6:3]ビットに記憶される。また、GSRは、以下に説明するように、シフトオペレーションから生じるビットを記憶するフィールドを設けることもでき、それによってビットがシフトしたことを示すか、または単にシフトが起こったことを示すフラグをセットする。GSR50からの読出し及びGSR50への書き込みを行うために、2つの特別な命令RDASR及びWRASRが用意されている。

【0023】FP／グラフィックスALU26

次に、図4には、装置26の第1の区分実行経路の一実施形態の関連部分を図解したブロック図が示されている。

【0024】パイプラインバス14は、PDU46からの復号された命令を2つの機能回路の1つに供給する。最初の2つの機能装置、すなわち区分桁上げ加算器37及びグラフィックス論理回路39は、浮動小数点加算器及び整数論理装置に通常に具備されるハードウェアを含む。回路はグラフィックス演算をサポートするよう変更されている。また、図解拡大及びマージオペレーションとグラフィックスデータ整列オペレーションを両方共をサポートするために別の回路60が付加されている。ライン21上の制御信号は、どの回路が復号された命令を受け取るかを選択し、また、どの出力がマルチプレクサ43を介して宛先レジスタ35cに供給されるかを選択する。宛先レジスタ35c、オペランドレジスタ35a及び35bは、図1の浮動小数点レジスタファイル38で特定のレジスタの例である。

【0025】各ディスパッチ毎に、PDU46は、グラフィックスデータ区分加算／減算命令、グラフィックスデータ整列命令、グラフィックスデータ拡大／マージ命令あるいはグラフィックスデータ論理演算命令を装置26へディスパッチすることができる。区分桁上げ加算器37は、区分グラフィックスデータ加算／減算命令を実行し、拡大／マージ／グラフィックスデータ整列回路60は、GSR50に記憶されたalignaddr_offsetを用いてグラフィックスデータ整列命令を実行する。また、グラフィックスデータ拡大／マージ／グラフィックスデータ整列回路60は、グラフィックスデータマージ／拡大命令も実行する。グラフィックスデータ論理演算回路39は、グラフィックスデータ論理演算

を実行する。

【0026】区分桁上げ加算器37の機能及び構成は、複数の加算/減算を区分されたオペランドの異なる部分に対して同時に行うことができるようにハードウェアを何回も重複させて設けた以外は、当技術分野で周知の多くの整数実行装置に見られる簡単な桁上げ加算器と同様である。さらに、桁上げチェーンは、任意により小さいチェーンに分解することができる。グラフィックスデータ論理演算回路39の機能及び構成は、複数の論理演算を区分されたオペランドの異なる部分に対して同時に行うことができるようにハードウェアを何回も重複させて設けた以外は、当技術分野で周知の多くの整数実行装置に見られる論理演算回路と同様である。従って、グラフィックスデータ論理演算回路39についてはこれ以上の説明しない。

【0027】F/P/グラフィックス乗算装置28

次に、図4には、F/P/グラフィックス乗算装置28の一実施形態の関連部分をさらに詳細に図解したブロック図が示されている。この実施形態においては、乗算装置28は、互いに図示のように接続されたピクセル距離計算回路56、区分乗算器58、グラフィックスデータバッキング回路59及びグラフィックスデータ比較回路64よりなる。さらに、いくつかのレジスタ55a~55c（浮動小数点レジスタファイル38中の）と4:1マルチプレクサ3が、図示のように相互に、また前記の各構成要素と接続されている。各ディスパッチ毎に、PDU46は、ピクセル距離計算命令、グラフィックスデータ区分乗算命令、グラフィックスデータバッキング命令、あるいはグラフィックスデータ比較命令を装置28へディスパッチすることができる。ピクセル距離計算回路56は、ピクセル距離計算命令を実行する。区分乗算器58は、グラフィックスデータ区分乗算命令を実行する。データバッキング回路59は、グラフィックスデータバッキング命令を実行する。グラフィックスデータ比較回路64は、グラフィックスデータ比較命令を実行する。

【0028】区分乗算器58及びグラフィックスデータ比較回路64の機能及び構成は、区分されたオペランドの異なる部分に対して複数の乗算及び比較演算を同時に行うことができるよう、ハードウェアを何回も重複させて設けた以外は、当技術分野の多くの整数実行装置で見られる単純な乗算器及び比較回路と同様である。さらに、区分乗算器には丸め機能用の複数のマルチプレクサが設けられ、比較マスクが比較回路64によって生成される。

【0029】本発明は、2つの独立した区分実行経路を有するグラフィックス回路の実施形態により、グラフィックス命令実行の特定の割当てをこれらの実行経路間で分担する場合について説明している。しかしながら、本発明のある部分を1つ以上の独立した区分実行経路を用

いて実施することができ、かつグラフィックス命令実行タスクは任意の数の方法で割り当てることができるということは理解できる。

【0030】データフォーマット

次に、図6の(A)及び(B)には、それぞれグラフィックスデータフォーマットとグラフィックス命令フォーマットが図解されている。図6(A)に示すように、図示例のCPU10は、3つのグラフィックスデータフォーマット、すなわち8ビットフォーマット(Pixel)66a、16ビットフォーマット(Fixed16)66b及び32ビットフォーマット(Fixed32)66cをサポートする。従って、32ビットワード66aには4つのピクセルフォーマットのグラフィックスデータが記憶されるのに対して、64ビットワード66bあるいは66cには4つのFixed16または2つのFixed32フォーマットのグラフィックスデータが記憶される。あるいは、8つのFixed8フォーマットのグラフィックスデータを64ビットワードに記憶することもできる。画像成分はPixelまたはFixed16フォーマット66aまたは66bに記憶される。標準的な音声データフォーマットもサポートされる。中間結果は、Fixed8、Fixed16またはFixed32フォーマット66bまたは66cで記憶される。あるいは、64ビット以上のフォーマットを含めて、その他任意のサイズのデータフォーマットを使用することができる。通常、画像のピクセルの輝度値、たとえばアルファ、緑、青、及び赤の値(α、G、B、R)は、ピクセルフォーマット66aで記憶される。これらの輝度値は、画像中の点の種々の色成分がまとめて記憶されるバンドインターリーブフォーマットで、あるいは1つの色成分についての全ての値がまとめて記憶されるバンド連続フォーマットで記憶することができる。Fixed16及びFixed32フォーマット66b、66cは、ピクセルデータに対して行われるフィルタリング及びその他の単純な画像処理オペレーションの間に計算される中間データを記憶するのに十分な精度とダイナミックレンジが確保される。

【0031】命令フォーマット

図6(B)に示すように、CPU10は3つのグラフィックス命令フォーマット68a~68cをサポートする。命令フォーマット68a~68cに関わらず、2つの最上位ビット[31:30]70a~70cは主要な命令フォーマット識別情報を与え、ビット[24:19]74a~74cはグラフィックス命令についての二次的な命令フォーマット識別情報を与える。さらに、ビット[29:25](rd)72a~72cは、グラフィックス(ブロック/部分的条件付きストア)命令の(第3のソース)宛先レジスタを識別するのに対して、ビット[18:14](rs1)76a~76cは、グラフィックス命令の第1のソースレジスタを識別する。

第1のグラフィックス命令フォーマット68aに関しては、ビット[13:5](opf)80及びビット[4:0](rs2)82aがそのフォーマットのグラフィックス命令に対する演算コード及び第2のソースレジスタを識別する。第2及び第3のグラフィックス命令フォーマット68b〜68cについては、ビット[13:5](imm asi)及びビット[13:0](simmm13)がそれぞれ任意にASI(アドレス空間識別子)を識別指示する。最後に、第2のグラフィックス命令フォーマット68bに関しては、ビット[4:0](rs2)がさらにそのフォーマットのグラフィックス命令のための第2のソースレジスタ(または部分的条件付きストアのためのマスク)を指示する。

【0032】論理演算

1. 乗算/加算(減算)

グラフィックス演算では、多数のピクセル値について、乗算に続けて加算または減算を行うことがしばしば必要になる。たとえば、乗法演算でピクセル値を一定値だけスケールリングしたり、またオフセット値を加えて3次元空間における位置を変えた場合がある。従って、本発明では、別個のオペランドを用いて乗算及び加算(または減算)の演算を両方とも行う単一命令を提供する。図7に示すように、乗算器90はレジスタ92及び94から入力を受け取る。レジスタ92は、たとえば多数の区分ピクセル値が書き込まれるソースレジスタであってもよい。レジスタ94は、たとえば、スケールファクタを記憶することができる。乗算の結果は、加算器/減算器96でレジスタ98からの値と加算される(この点が、インテルMMX命令で行われるように、乗算結果の区分フィールドが互いに加算されるのと異なる)。レジスタ98中の値は、たとえば、オフセットであってもよい。

【0033】命令フォーマットの一例においては、図6の(B)のフォーマット68aをレジスタ92中の区分ピクセル値を指示するRD、レジスタ94のスケールファクタを指示するRS1及びレジスタ98のオフセット値を指示するRS2で使用する事ができる(1つのレジスタ、すなわちRDがソース及び宛先の両用に使用されたとすることに留意すべきである)。

【0034】演算の結果はRDによって指定される宛先レジスタに記憶される。各ピクセル値は、乗算後宛先レジスタの対応するフィールド内に適合させるように切り捨てられ、または飽和させることができる。

【0035】マスクレジスタ95は、3つのオペランドのいずれかあるいは乗算器90の中間出力中の指定された区分フィールドをマスクするために使用することができる。

【0036】好ましくは、中間乗算結果については丸めは行わない。これによれば、2命令方式と比較して1つの丸め段が省かれ、余分な実行時間が節約される。

【0037】2. 平方根で割る1

図形演算では、ある数の平方根を求め、次にその逆数(1/X)を計算することが必要なことがしばしばある。たとえば、グラフィックス演算で用いられる多くの三角関数ではこれが必要である。Xは通常ピクセル値またはピクセルアドレスである。通常、平方根演算は、除法演算と同様、所望の精度に応じた演算を行うのに適切な論理回路を介しての複数回の反復パスを必要とする。しかしながら、バククされたピクセルフォーマットを使用する場合は、分割する、あるいは平方根を計算する各ピクセルについて、ビット数は限定されたものになる。従って、ただルックアップテーブルを使うだけでピクセル値の平方根分の1に等しい値を得ることが可能である。そのようなルックアップテーブルを図8の(A)にテーブル100として示しており、このテーブルは、出力102にピクセル値の平方根分の1の値を出力する。入力は、バス106を介してソースレジスタ104から供給される。このテーブルは並列に複数の出力を供給するよう構成することができ、あるいはレジスタ104からの区分値をルックアップテーブルに逐次供給し、その後結果を逐次宛先レジスタの適切なフィールドに入力することも可能である。あるいは、2セットの反復演算で除算と平方根演算を別個に行うのと比較して時間/節約される複合演算のための1セットの反復による反復演算を用いることも可能である。

【0038】3. A+ABS, [B]

グラフィカルアプリケーションでは、ピクセル値と絶対値との結合を計算したいことがしばしばある。たとえば、これはオペレーション推定及び検出に利用される。この演算は、ソースレジスタ中の複数の区分ピクセル値について並列に実行される。絶対値を計算したり、第2のオペランドの2の補数を取るための論理は第2のオペランドの符号ビットによって決まる。

【0039】図8の(B)は、ある値と第2の値の絶対値との加算を実行するための論理回路の一例を示したものである。図示の論理回路は、たとえば区分ピクセルフィールドの1つについてのものであり、たとえば各ピクセルフィールドについて反復して設けられる。加算器101は、レジスタRS1(103)から値AとレジスタRS2(105)からBの絶対値を受け取り、加算結果はRD宛先レジスタ107に供給される。Bの値は2の補数論理回路109によってその絶対値に変換される。

【0040】絶対値演算は、マルチプレクサ113及び115を制御する演算コード111を復号することによって起動される。この演算が普通の加算ならば、マルチプレクサ113及び115の“0”入力を選択される。この演算が普通の減算であれば、マルチプレクサ115の“1”入力とマルチプレクサ113の“0”入力を選択される。絶対値が加えられる場合は、マルチプレクサ113の“1”入力を選択される。RS2符号ビット119は、ライン119上の区分フィールドについてのR

S2符号ビットの値に応じて1またはゼロになる。

【0041】データ移動オペレーション

1. 区分フィールド抽出

多くのグラフィックスアプリケーションでは、指定されたピクセルを選んで移動させたり、あるいは処理したりできるようにすることが望ましい。ピクセルは複数のピクセルが1つのレジスタに入るようにしてバックされるので、標準的なオペレーションでは、ピクセルがアンバックされない限り、これらの移動や処理は達成されない。本発明によれば、フィールドをソースから宛先レジスタへ選択的に移動させ、そのようなフィールド中のデータに選択的に演算を加える命令及び論理が提供される。図9の(A)に示すように、複数フィールドを有するソースレジスタ108は、マスクレジスタ112によって指示される指定フィールドを宛先レジスタ114へ送るマルチプレクサネットワーク110に接続されている。

【0042】図9の(B)は、文字A、B、C及びDがソースレジスタ108中のピクセル値を指示する1つの例を示したものである。マスクレジスタは値1010が書き込まれており、“1”の値はそのフィールドを宛先レジスタ114に送るべきであるというを示す。図から明らかなように、“1”の値はピクセル値BとDに対応しており、これらは宛先レジスタ114の最下位位置へ送られる。

【0043】移動命令に加えて、ピクセル値は、この方法により選択的にメモリからレジスタにロードすることもできる。さらに、ピクセル値は、この方法により選択的に演算（乗算または加算のような）を加えることができる。

【0044】選択されたピクセルに対して演算を加えるための命令は、たとえば2つの演算コードで実行することができる。その第1の演算コードは、たとえばマスク値をセットするものであり、第2の演算コードは、たとえば移動・加算オペレーションを指定するものであり、第1のレジスタはソースレジスタとして指定され、第2のレジスタではソースレジスタからの選択された各ピクセル値に加えられる値が指定される。

【0045】図9の(A)及び(B)は簡単な抽出命令の例を示したものであるが、図13には、マスクレジスタを用いると共に、個々のフィールドに対して随時算術演算または論理演算を行って、特定のフィールドの選択を行う例が図解されている。図13に示すように、ソースレジスタ108の内容は論理回路116を介して宛先レジスタ114へ供給される。マスク112は、たとえば加算オペレーションを行うことができる116で示す論理ブロックをイネーブル、またはディスエーブルにする。あるいは、マスクによって指定された宛先レジスタの各部分の動作をディスエーブルにするようにしてもよく、あるいはその他任意のマスキング機構を使用するこ

とが可能である。図13の実施形態においては、選択されたピクセル値は、図9(B)の実施形態におけるように最下位フィールドにバックされるのではなく、宛先レジスタの対応する位置に供給される。

【0046】図9の(C)は、条件付きマージ動作を図解したものである。図示のように、レジスタ114の各部分はレジスタ108の部分とマージされ、マスク112はレジスタ108のどの区分フィールドがレジスタ114のフィールドを上書きするかを指示する。上書きされないレジスタ114のフィールドは、そのままの状態に保たれる。

【0047】2. 浮動小数点グラフィックスレジスタファイル及び整数レジスタファイル交換

図11は、整数レジスタファイル36と浮動小数点/グラフィックスレジスタファイル38の間でデータを交換する命令を実行するための論理構成を示す。制御論理回路118は、データを転送するためのバッファ120及び122をイネーブルにするよう動作する。バッファ120は、整数レジスタファイルへ転送される浮動小数点/グラフィックスレジスタファイルからレジスタ124のデータ内容をバッファリングするために用いられる。同様に、バッファ122は、浮動小数点グラフィックスレジスタファイル38へ転送される整数レジスタファイル36からレジスタ126の内容を一時的に記憶する。2つのレジスタの内容をスワッピングする他、命令によって、1つのレジスタの内容を単に空いているレジスタに移動するか、その内容で他方のレジスタファイルのもう一つのレジスタを上書きするようにすることも可能である。この操作によれば、計算を1つのレジスタファイルで行い、その結果が他方のレジスタファイルに必要となるような演算の場合に、メモリに書き込んだ後、次にそのメモリから別のレジスタファイルロードする必要があることになる。たとえば、アドレスを浮動小数点/グラフィックス実行装置を用いて計算し、その結果を浮動小数点/グラフィックスレジスタファイルに記憶することがある。その場合、整数実行装置のアドレスを使用することが望ましいことがあり、この操作を用いて転送を行うことができる。

【0048】たとえば、レンダリング操作にレジスタファイルの間のスワッピングが必要なことがある。加算または減算する値を、アドレス計算のためのオフセットとして使用するためにロード及びストア動作によってアクセスすることができるように、浮動小数点レジスタファイルから整数レジスタファイルに移動することが必要な場合もある。

【0049】3. 区分シフト

図12は、区分シフト動作をサポートするための論理構成を示したものである。この場合、単一レジスタ中の複数のピクセル値は各々それらの区分フィールド内でシフト、すなわち桁送りされる。ソースレジスタ130は、

シフト論理回路132に区分フィールドを提供し、論理回路の結果は宛先レジスタ134の対応する区分フィールドに入れられる。シフトカウンタ136はシフト量を決定する。あるいは、シフト量は演算コードから埋め込むか、または暗黙で得ることもでき、あるいはGSRレジスタのフィールドに記憶することもできる。矢印138示すように、値0が左シフトによって各区分フィールドに入れられる。一つの選択肢として、シフトアウトされたビットはマスクまたは制御レジスタ140に供給するようにしてもよい。レジスタ140は、たとえば、シフトが起ったことを指示するフラグをセットするために使用することができる。あるいは、マスク140は、どの区分フィールドをシフトさせるかを破線の制御ライン141を介して選択するのに使用される。

【0050】論理演算あるいは算術演算においては、右シフト動作を行うこともできる。算術演算の場合、符号ビットは、ビットがシフトされるにつれて繰り返し挿入されるようにしてもよい。

【0051】メモリアクセス動作

1. ロードとアドレスインクリメント。

本発明では、やはりアドレスレジスタをインクリメントするロード動作が用いられる。これによって、アドレスレジスタをインクリメントさせるための別の命令を用いる必要がなくなる。このことは、グラフィックス演算は、事実上大量のデータによって進行し、繰り返しインクリメントが必要になることが多いので、非常に重要である。ロードは、好ましくはグラフィックス/浮動小数点レジスタファイルのグラフィックスレジスタに対して行われる。ロードは、データサイズに応じて決めることが可能な適切なアドレスインクリメントを指定することによる複数の区分フィールドを含んでもよい。レジスタ全体（たとえば、64ビット）を一度にロードすることもできる、は1または複数の区分フィールドをロードするようにしてもよい。

【0052】図14は、ロード・インクリメント命令をサポートするための回路の一実施形態を示す。図示のアドレスレジスタ142は、ライン144を介してメモリ146にアドレスを供給する。メモリ146からのアドレス指定されたデータは、入力ライン148（144と同じバスであってもよい）を介してグラフィックス宛先レジスタ150に供給される。さらに、加算器152がアドレスレジスタ144の入力からその出力を供給することによってインクリメント・オペレーションが行われ、そのインクリメントの大きさはレジスタ154中の値によって指示される。

【0053】上記実施形態は、コンピュータのハードウェアによって実施される。そのハードウェアシステムで用いられるプログラムは当然のことながら記録媒体、すなわちメモリに記録された状態で提供される。このプログラムを記憶させたメモリとしては、例えばフレイグ

ルディスク、CD-ROM、メモリアカードその他あらゆるメモリを使用できる。メモリに記録されたプログラムは、ハードウェアに組み込まれている記憶装置、例えばハードディスクなどにインストールすることにより、プログラムが実行できるようになる。

【0054】当技術分野の当業者には容易に理解できるように、本発明は、その精神あるいは必須の特徴から逸脱することなく他の特定の形態で実施することが可能である。従って、前記実施形態は例示説明を目的とするものであり、特許請求の範囲に記載する本発明の範囲を限定するためのものではない。

【図面の簡単な説明】

【図1】 本発明の技術内容を組み込んだグラフィックスコンピュータシステムの一実施形態のCPUを示すブロック図である。

【図2】 図1で付加されたグラフィックス回路の一実施形態の2つの区分実行経路を示すブロック図である。

【図3】 グラフィックス状態レジスタ（GSR）の説明図である。

【図4】 図2の第1の区分実行経路（ALU）をより詳細に示すブロック図である。

【図5】 図2の第2の区分実行経路（乗算）をより詳細に示すブロック図である。

【図6】 グラフィックスデータフォーマット及びグラフィックス命令フォーマットをそれぞれ（A）及び（B）に示す説明図である。

【図7】 乗算と加算の複合演算を行うための論理構成を示す論理図である。

【図8】 平方根による除算及び $A+ABS[B]$ を可能にする論理構成をそれぞれ（A）及び（B）に示す論理図である。

【図9】 特定の区分フィールドからのデータの選択抽出及び条件付きマージ動作を（A）、（B）及び（C）に示す説明図である。

【図10】 先行1またはゼロを検出するための構成2つの実施形態を（A）及び（B）に示す説明図である。

【図11】 整数レジスタファイルと浮動小数点/グラフィックスレジスタファイルとの間のレジスタ内容のスイッチングを図解した説明図である。

【図12】 区分シフト論理を図解した説明図である。

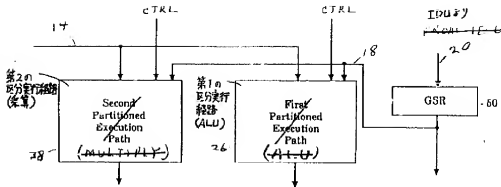
【図13】 特定の区分フィールドの選択移動のための論理構成を図解した説明図である。

【図14】 ロードとアドレスインクリメントの複合命令を実行するための論理構成を図解した論理図である。

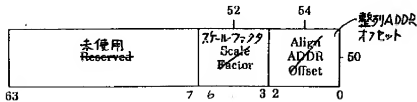
【符号の説明】

12 バス、14 バイプレインバス、30 整数乗算/除算装置、31 整数ALU、36 整数レジスタファイル、40 命令キャッシュ、44a 命令メモリ管理装置、44b データメモリ管理装置。

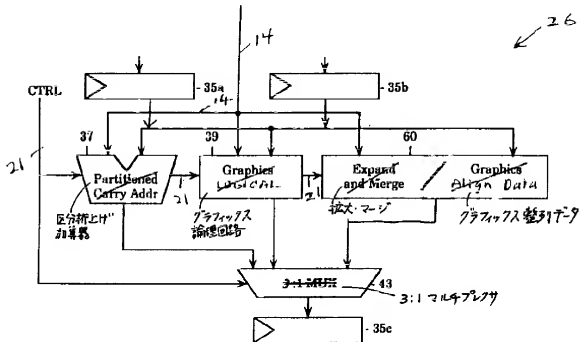
【図2】



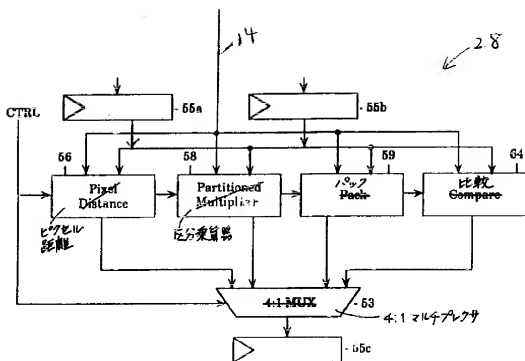
【図3】



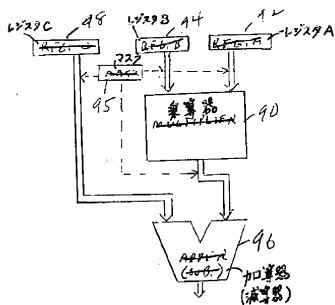
【图4】



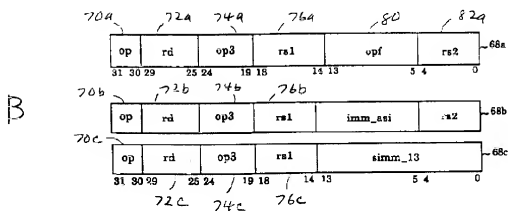
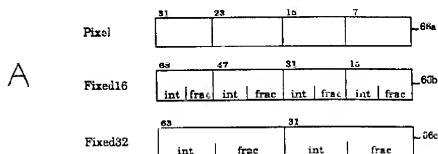
【図5】



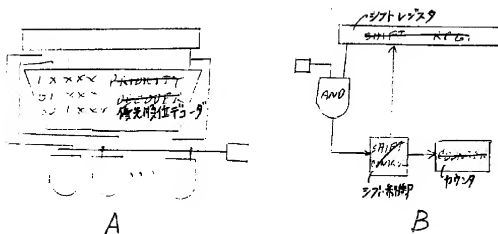
【図7】



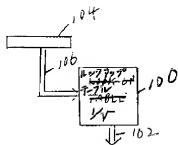
【図6】



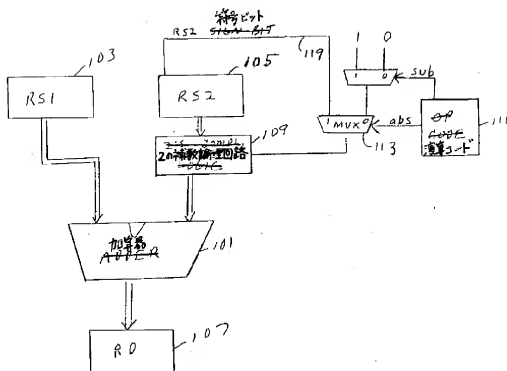
【図10】



【例8】

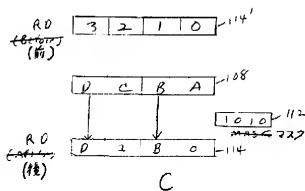
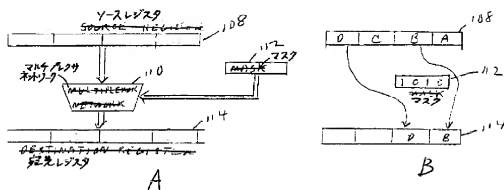


A

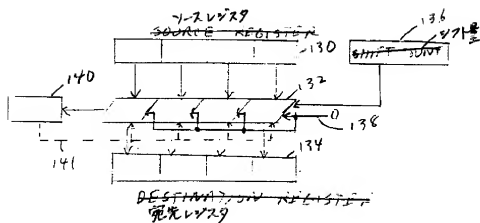


B

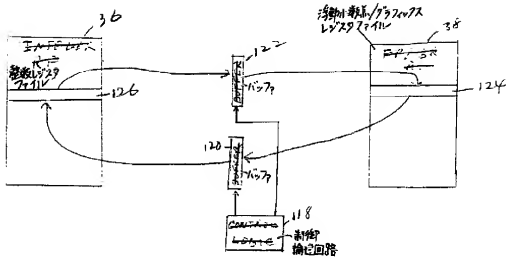
【図9】



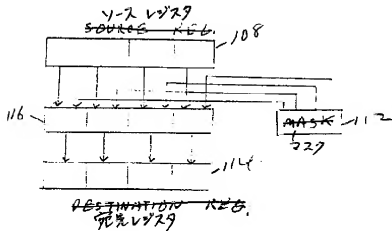
【図12】



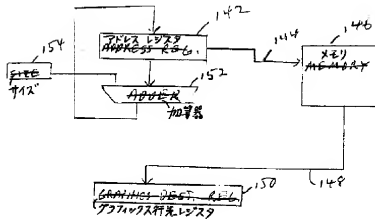
【図11】



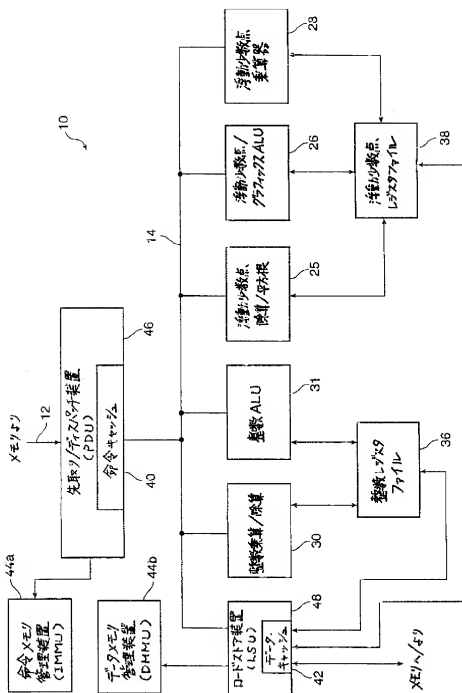
【図13】



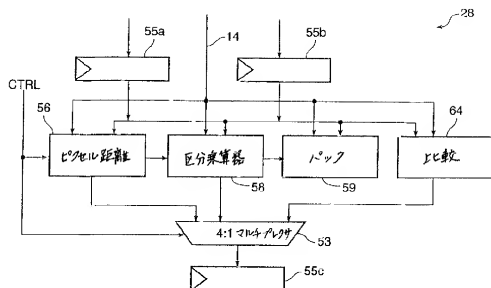
【図14】



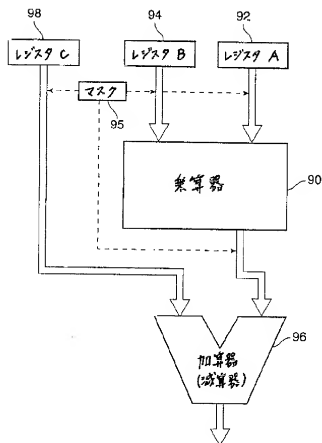
【図1】



【図5】

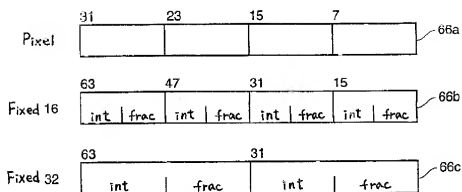


【図7】

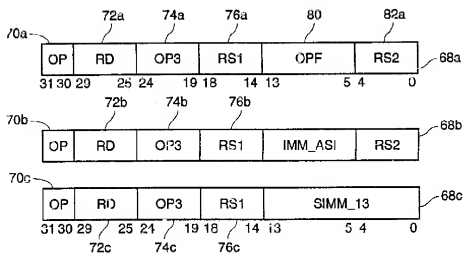


【図6】

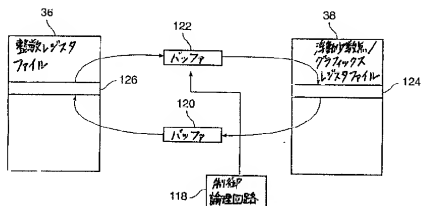
A



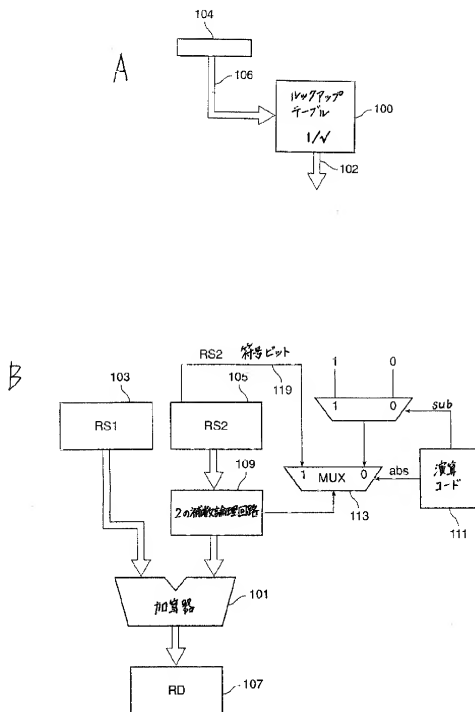
B



【図11】



【図8】



【図9】

